How to generate algorithm

Algorithm to compute the network reliability, using the method of exhaustive enumeration:

1. How to generate the possible states.

2. How to assign an up/down system condition to each

3. How to convert it into a reliability value

\* Describe how algorithm works

\* Provide pseudo code

\* With sufficient comments

Pseudo Code:

1. we are going to calculate all possible combinations, since 10 edges will have 210 combinations, we store them in a Boolean arry: Boolean[] combinations = new Boolean[210]

ArrayList<Boolean[]> combinations= new ArrayList<Boolean[numOfEdges]>();

Up = 1, Down = 0.

when we want kth combination, we get the kth ArrayList.

setCombinations(int numOfEdges, List combinations){

combinations.add(false,false,false,false,false,false,false,false false,false);

for(i = 0; i < numOfEdges; i ++){

Boolean temp[] = new Boolean[10];

int j = i;

while(j >= 0){

temp[j--] = true;

}

for(j = i + 1; j < numOfEdges; j++){

}

}

}

combine(int start, int end, Boolean temp[]){

if(start == numOfEdges)

return;

temp[start] = true;

for(int i = start + 1; i < numOfEdges; i++){

}

}

Exhaustive enumeration.

List all possible states of the system.

Assign “up” and “down” system condition to each state

Reliability can be obtained by summing the probability of the “up” states

for 5 nodes 10 edges, we assume nodes all work well, each edge is either Up or Down, and we will know the p for each edge.

if the system is operational, it means there is no unconnected node

Tasks:

1. Checking correctness

2. Explain how your program supports these goals

3. ReadMe file

4. Run for different value of p, let p run over [0.05,1] in steps of 0.05, show how the obtained network reliability values depend on p

5. Then fix p = 0.9, among 210 combination states pick k of combinations randomly and flip the corresponding system condition, show in diagram how the reliability of the system changes due to this alteration. Show in diagram how the change depends on k range 0,1,2,3….20, run several experiments for each k, and average them out. Give several paragraph explanation.

Task 4:

Rsystem = 1 – anyNodeNotWork

= 1 – (1 – allNodeWork)

= 1 – (1 - oneNodeWorknumOfNodes)

= 1 – (1 - (1 – allEdgesOfNodeNotWork)numOfNodes)

= 1 – (1 - (1 – (1 - p)numOfNodes - 1 )numOfNodes)